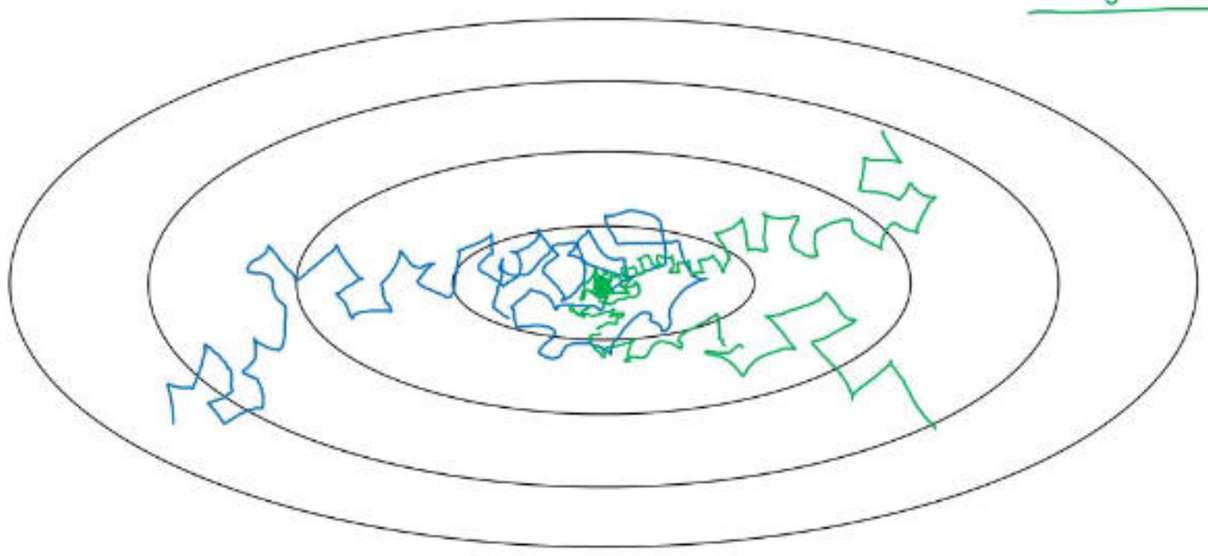
日期: 12 /20

Learning rate Decay.

Learning rate decay



Andrew Ng

从随 epoch↑ 逐渐减小，在最小值的小邻域内波动.

常见公式: $\alpha = \frac{1}{1 + \text{decay-rate} * \text{epoch}} \alpha_0$

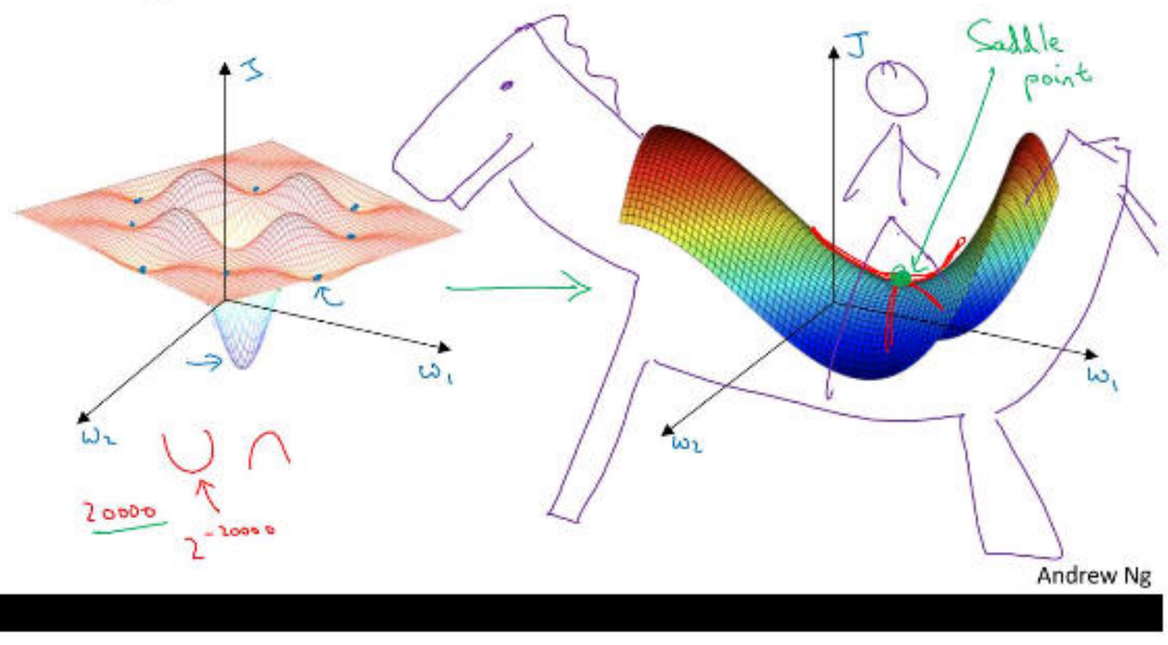
$\alpha = 0.95^{\text{epoch}} * \alpha_0$

$\alpha = \frac{k}{\sqrt{\text{epoch}}} * \alpha_0$ or $\frac{k}{\sqrt{t}} * \alpha_0$

Local optima.

高维空间不容易出现最部最优，多为鞍点.

Local optima in neural networks



Andrew Ng

鞍点减缓学习过程，可以通过 mini-batch 扰动摆脱，很慢。。。

日期: 12 / 20.

Hyper Param Tune.

Prior:

1. α 2. β, # hidden-units, batch size. → 加权平均.

3. # layers, learning rate decay,

4. β_1, β_2, ε.

0.9 0.999 10^{-8}.

调参方法: 1. 随机采样 2. 由粗到精.

3. baby sit one model ~ Panda.

Train models in parallel ~ Caviar.

Batch 归一化:

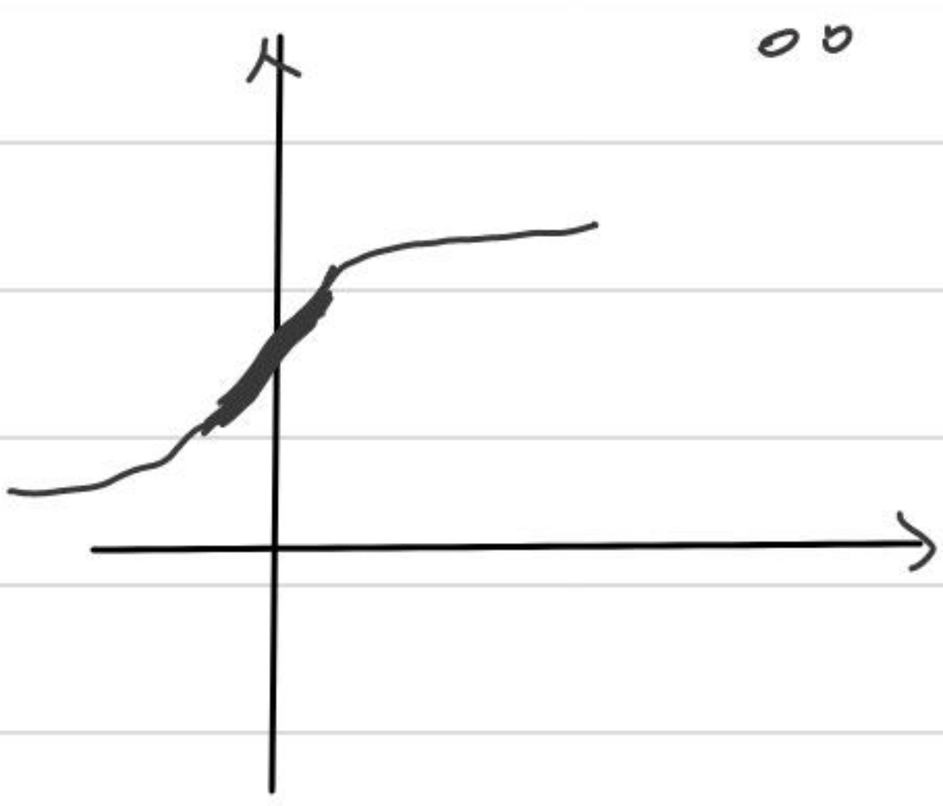
对于 x_i, 做归一化.

类似地, 对 $z^{[l](1)} \cdots z^{[l](m)}$ 也做归一化.

$$\mu = \frac{1}{m}\sum_i z^{(i)} \qquad \sigma^2 = \frac{1}{m}\sum_i (z^{(i)} - \mu)^2$$

$$z^{(i)}_{norm} = \frac{z^{(i)} - \mu}{\sqrt{\sigma^2 + \varepsilon}}. \qquad \tilde{z}^{(i)} = \gamma z^{(i)}_{norm} + \beta.$$

β, γ 用于调整方差与均值.



不希望 $z^{[l]}$ 总是输入线性激活函数.

日期: 12 / 21

应用到 NN:

$$z^{[l]} \xrightarrow{\beta^{[l]},\ \gamma^{[l]}} \tilde{z}^{[l]} \longrightarrow a^{[l]} \longrightarrow \cdots$$

$\beta^{[l]}$, $\gamma^{[l]}$ 也视作参数进行 Gradient Descend.

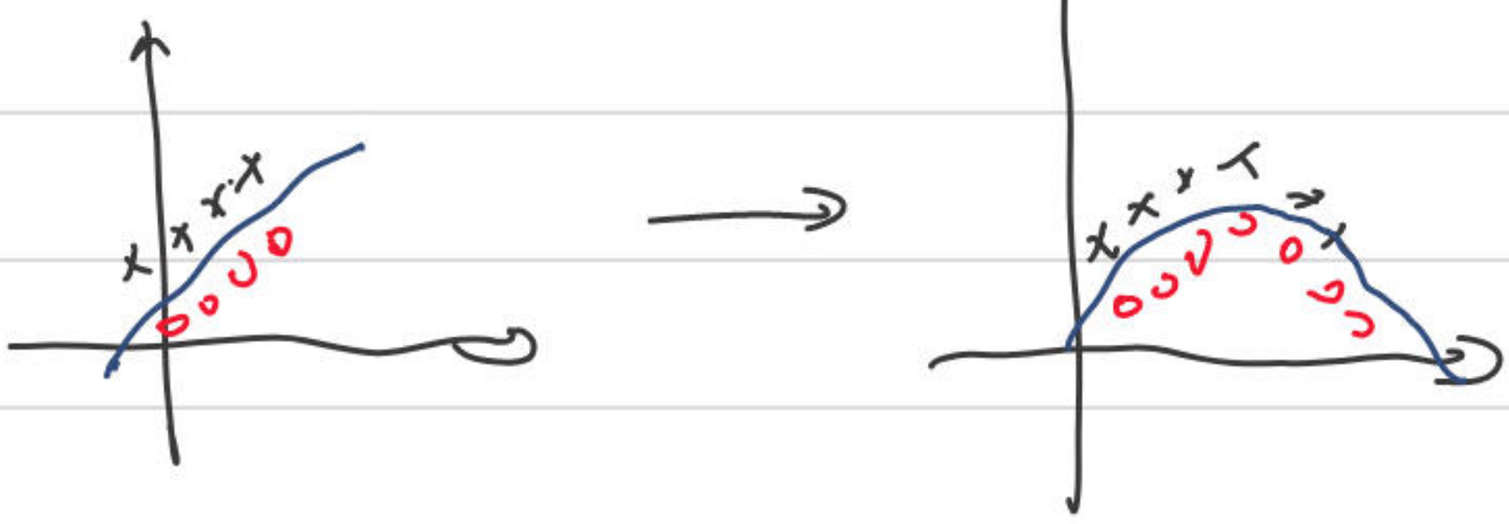
$z^{[l]} = w^{[l]} a^{[l-1]} + ~~b^{[l]}~~$ $\qquad A^{[l]} = g(z^{[l]})$.

$\tilde{z}^{[l]} = \gamma^{[l]} z^{[l]}_{norm} + \beta^{[l]}$

Params: $w^{[l]}$, $\beta^{[l]}$, $\gamma^{[l]}$.

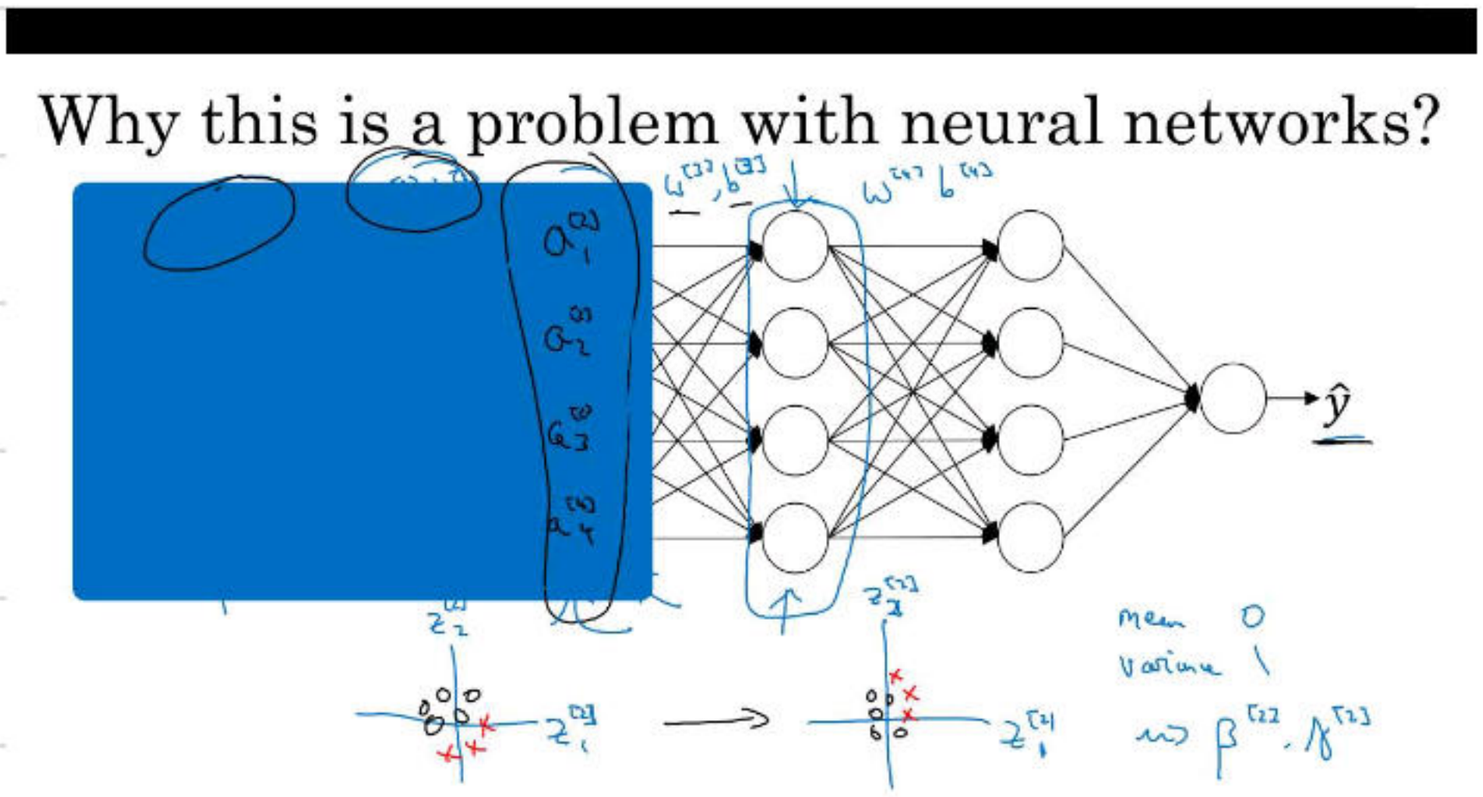
$(n_l \times 1)$

Intuition:



Data 改变了, 应重新训练 Model,

$a^{[l]}$ 因 变动, Model 重新训练, 归一化使 变动变小, 加速训练.



日期: 12 / 21.

μ, σ^2 是在 batch 上计算的, 有噪声, 有一定 Reg 效果.

测试时样本不多, 不就算 μ, σ^2.

在训练集上对 $\mu^{[l]}$ 做指数加权平均.

Soft Max.

C = # classes. $n_l = 4$. $a^{[l]} = (n_l, 1)$.

$z^{[l]} = w^{[l]} a^{[l-1]} + b^{[l]}$.

激活函数: $t = e^{z^{[l]}}$ $z^{[l]} = (4,1)$, $e^{z^{[l]}}$ $(4,1)$.

$$a_i^{[l]} = \frac{t_i}{\sum_j t_j}$$

Hard Max ↓ VS Soft Max. ↓

[1 0 0 0] [0.8 0.1 0.1]

若 C=2, Soft Max $\Leftrightarrow$ Logistic.

$$L(\hat{y}, y) = -\sum_{j=1}^{4} y_i \log \hat{y}_j$$

$$J(w^{[l]}, b^{[l]}) = \frac{1}{m} \sum_{i=1}^{m} L(\hat{y}^{(i)}, y^{(i)})$$

Back Prop: $\frac{\partial J}{\partial z^{[l]}} = \hat{y} - y$.

日期: 12/21

TensorFlow.

通过前向传播绘制计算Graph. 自动反向传播.